Optimal Joint Routing and Scheduling in Millimeter-Wave Cellular Networks

Dingwen Yuan*, Hsuan-Yin Lin†, Jörg Widmer‡ and Matthias Hollick*

*SEEMOO, Technische Universität Darmstadt   †Simula@UiB, Bergen Norway   ‡Institute IMDEA Networks

{firstname.lastname}@seemoo.tu-darmstadt.de   hsuan-yin.lin@ieee.org   joerg.widmer@imdea.org

Abstract—Millimeter-wave (mmWave) communication is a promising technology to cope with the expected exponential increase in data traffic in 5G networks. mmWave networks typically require a very dense deployment of mmWave base stations (mmBS). To reduce cost and increase flexibility, wireless backhauling is needed to connect the mmBSs. The characteristics of mmWave communication, and specifically its high directionality, imply new requirements for efficient routing and scheduling paradigms. We propose an efficient scheduling method, so-called *schedule-oriented optimization*, based on matching theory that optimizes QoS metrics jointly with routing. It is capable of solving any scheduling problem that can be formulated as a linear program whose variables are link times and QoS metrics. As an example of the schedule-oriented optimization, we show the optimal solution of the maximum throughput fair scheduling (MTFS). Practically, the optimal scheduling can be obtained even for networks with over 200 mmBSs. To further increase the runtime performance, we propose an efficient edge-coloring based approximation algorithm with provable performance bound. It achieves over 80% of the optimal max-min throughput and runs 5 to 100 times faster than the optimal algorithm in practice. Finally, we extend the optimal and approximation algorithms for the cases of multi-RF-chain mmBSs and integrated backhaul and access networks.

Fig. 1. mmWave self-backhauling setup.

I. Introduction

5G cellular systems are embracing millimeter wave (mmWave) communication in the 10-300 GHz band where abundant bandwidth is available to achieve Gbps data rates. One of the main challenges for mmWave systems is the high propagation loss at these frequency bands. Although it can be partially compensated by directional antennas [1], [2], the effective communication range of a mmWave base station (mmBS) remains around 100 meters at best. Thus, base station deployment density in 5G will be significantly higher than in 4G [3], [4]. This leads to high infrastructure cost for the operators. Besides the cost of site lease, backhaul link provisioning is in fact the main contributor to this cost because the mmWave access network may require multi-Gbps backhaul links to the core network. Currently, such a high data rate can only be accommodated by fiber-optic links which have high installation cost and are inflexible with respect to relocation.

Recent studies show that mmWave self-backhauling is a cost-effective alternative to the wired backhauling. This approach is particularly interesting in a heterogeneous network setting where the existing cellular base stations (eNBs) act as a gateway for the mmBSs. Fig. 1 illustrates such a setup in which the eNB can reach mmBSs directly or via other 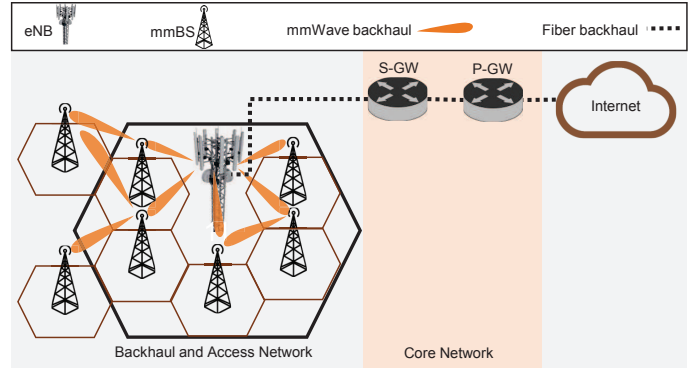mmBSs. Moreover, directionality of mmWave communication reduces or removes the wireless backhaul interference and allows simultaneous transmissions of multiple links over the same channel as long as their beams do not overlap. However, the number of simultaneous links a base station can have is limited by the number of its RF chains.

To date, much of the research on mmWave communication has been dedicated to issues faced by the mobile users (UEs) in the access networks. How to maximize performance such as throughput and energy efficiency in mmWave backhaul and access networks has received less attention. Here, two important issues need to be addressed: (i) routes to be taken, (ii) the scheduling of the transmission overs the links.

A naive scheduling which lets the eNB serve all the mmBSs in a round robin fashion is neither practical nor efficient. If mmBSs' links to the eNB are weak compared to their links to other nearby mmBSs (which in turn have high-capacity links to the eNB), a schedule allowing multi-hop routing is much more favorable since it alleviates the bottleneck at the eNB. At the same time, the limited interference at mmWaves makes it efficient to maximize spatial reuse and operate as many links simultaneously as possible. The goal of the paper is to design a scheduler that exploits these characteristics to optimize mmWave backhaul efficiency.

The paper is organized as follows. We discuss related work in Sec. II. Sec. III provides the system model. The relation between a schedule and matchings is studied in Sec. IV. In Sec. V, we present our optimal schedule-oriented optimization method, through an example—maximum throughput fair scheduling. We then propose a fast edge-coloring based

approximation algorithm in Sec. VI. In Sec. VII, we extend our algorithms to more general scenarios. Sec. VIII shows the numerical evaluation and Sec. IX concludes the paper.

II. RELATED WORK

Few works on mmWave backhaul and access network scheduling exist [5]–[8]. These works share the assumptions that (i) the traffic demand is measured in discrete units of slots or packets, and (ii) a flow has to be scheduled sequentially, i.e., a hop closer to the source should be scheduled earlier than a hop farther away. The resulting optimization problems are all formulated as mixed integer programming (MIP) problems. As MIPs are in general NP-complete, optimal solutions can only be computed for small networks with a few nodes. For practical use, these works all rely on heuristics, which are based on the ideas such as greedy edge coloring [5], [7] or finding the maximum independent set in a graph [6], [8]. Furthermore, [5], [6], [8] assume that routing is pre-determined, which does not fully exploit the freedom given by a reconfigurable mmWave backhaul, and may limit performance.

In contrast, our work relaxes the constraint of sequential flow scheduling (i.e., if needed, packets are queued for a short time) which does not harm the long-term throughput, and allows the slots in a schedule to be of any length. Based on these assumptions, we propose a polynomial time optimal scheduling method which is shown by simulation to be practical for mmWave cellular networks. Moreover, the scheduling takes QoS optimization goals or QoS requirements as input and finds an optimal routing automatically. The first attempt to solve the problem of joint routing and scheduling in a network with Edmonds' matching formulation goes back to [9]. Hajek et al.'s polynomial time scheduling algorithm is different from ours in that it minimizes the schedule length. Furthermore, we use a one-step *schedule-oriented* approach while they first compute the optimal link time and then compute the minimum length schedule given the link time.

Following [9], recent research on scheduling focuses on optimization of delay [10] and queue length [11], as well as investigating more realistic interference models [12]. Another interesting line of research is the on-line node-based scheduling algorithms which achieve good performance bound in throughput and evacuation time [13].

III. SYSTEM MODEL

The system model considers a backhaul network which has a single eNB, equipped with multiple mmWave RF chains, and multiple mmBSs, each equipped with a single RF chain. Later in Sec. VII, we will show that our optimization method applies equally to (i) the case where each node has multiple RF chains, and (ii) a network model that includes UEs.

We consider an eNB macro cell together with a number of mmWave base stations in a heterogeneous TDMA cellular network. The eNB acts as the backhaul gateway for mmBSs. In addition to an LTE radio interface, the eNB is equipped with R mmWave RF chains. There are W single-RF-chain mmBSs in the macro cell. We assume analog or hybrid beamforming with R RF chains which allows up to R simultaneous links at the eNB. We use directed graphs to model the links between the
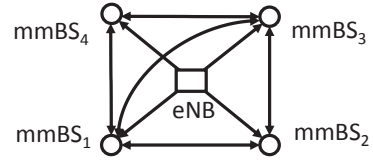

Fig. 2. A backhaul network example. The edges shown are the potential links for downlink schedule. A bidirectional edge represents two opposite links.

different nodes in the network. Fig. 2 illustrates a toy example of a backhaul network. The following analysis focuses on downlink communication. The same analysis can be applied to the uplink scenario.

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with the vertex (node) set $\mathcal{V}$ and edge (link) set $\mathcal{E}$. Each edge represents a potential link between two vertices. The capacity of each edge e is denoted c_e. The received power is given by $p_{\text{rx}} = p_{\text{tx}} + g_{\text{x}} - PL$, where p_{tx} is the transmission power, g_{x} is the directivity gain, and PL is path loss between the transmitter and the receiver. $PL(d) = \alpha + 10\beta \log_{10} d + \xi$, where α, β are constants that depend on the frequency and line-of-sight conditions. d is the distance between the transmitter and receiver. ξ represents the shadowing effect and is a normal distributed random variable with zero mean and σ standard deviation.

We can observe in Fig. 2 that there are many ways to schedule downlink communication among the eNB and mmBSs. Our goal is to obtain the optimal *unit length schedule* with respect to a QoS metric, while satisfying given QoS requirements and the constraints on simultaneous transmissions. In practice, the unit time is the duration of the radio frame. We observe that each feasible schedule $\mathcal{S}$ can always be divided into $N \geq 1$ slots numbered as 1 to N. We define t_i as the length of the i-th slot. It is required that $\sum_{i=1}^{N} t_i = 1$ and $t_i > 0$. Moreover, in the i-th slot, a set of links $\mathcal{E}_i \subseteq \mathcal{E}$ (can be empty) are active for the whole slot.

IV. PRELIMINARY: SCHEDULE POLYHEDRON

This section first shows the relation between a feasible schedule and matchings in a graph. Based on the relation, we mathematically formulate the set of all feasible schedules as the *schedule polyhedron* that is described by linear constraints.

Suppose that a set of links $\mathcal{E}_i$ are scheduled in the i-th slot, and $e_1, e_2 \in \mathcal{E}_i$ are two different links. Then e_1 and e_2 can not share a common mmBS node since a mmBS has one RF chain and is therefore half-duplex. On the other hand, e_1 and e_2 may share the eNB node given that $R > 1$. However, the number of links in $\mathcal{E}_i$ that are incident to the eNB cannot be more than R. We enforce this constraint through the eNB expansion.

A. eNB expansion

In graph G, we replace the eNB with R *expanded eNBs*: $\text{eNB}_1, ..., \text{eNB}_R$. If the eNB is connected to a set of mmBSs, then each eNB_i is connected to the same set of mmBSs with the same respective link capacities as the eNB. The resulting graph is equivalent to the original graph with respect to scheduling. Yet each expanded eNB has one RF chain. In the rest of the paper, the graph $G = (\mathcal{V}, \mathcal{E})$ is assumed to be
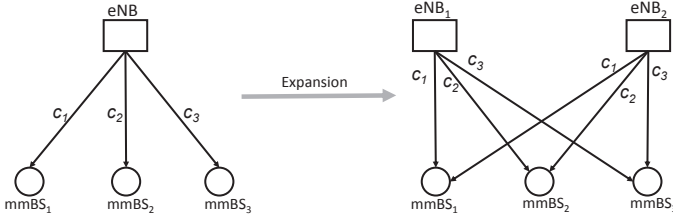
Fig. 3. eNB expansion with two RF chains.

Fig. 4. Node-matching matrix, a to d are nodes, α to ϵ are edges. The numbers are capacities.

expanded if not explicitly stated otherwise. An example of the process described above is shown in Fig. 3. Let $\mathcal{R}$ and $\mathcal{W}$ denote the set of expanded eNBs and mmBSs, respectively.

As a result, we can ensure that the active link set of a slot corresponds to a *matching* in the expanded graph. A matching in a graph is defined as a set of edges in the graph that share no common vertices.

B. The schedule polyhedron

We define the *link time* $t_e \in [0,1]$ as the total active time of a link e in a schedule. Correspondingly, $\boldsymbol{t}$ is the *link time vector*, each element of which is a link time $t_e, \forall e \in \mathcal{E}$. A link time vector $\boldsymbol{t}$ is *feasible* if $\boldsymbol{t}$ can be scheduled in unit time. We first define the *schedule polyhedron* P and then prove that each point in P is one-to-one mapped to each feasible link time vector.

Definition 1 (Schedule Polyhedron). *Given a graph $G = (\mathcal{V}, \mathcal{E})$, the schedule polyhedron of G is defined as the set of link time vectors $\boldsymbol{t}$ that satisfy the following linear constraints.*

$$\sum_{e \in \delta(v)} t_e \leq 1 \qquad \forall v \in \mathcal{V}, \tag{1a}$$

$$\sum_{e \in \mathcal{E}(\mathcal{O})} t_e \leq \left\lfloor \frac{|\mathcal{O}|}{2} \right\rfloor \qquad \forall \text{ odd set } \mathcal{O} \subseteq \mathcal{V}, \tag{1b}$$

$$t_e \geq 0 \qquad \forall e \in \mathcal{E}, \tag{1c}$$

where $\delta(v)$ is the set of links incident to node v. An odd set *$\mathcal{O}$ has odd number of nodes. $\mathcal{E}(\mathcal{O})$ is the set of links whose endpoints are both contained in $\mathcal{O}$.*

The following lemma summarizes the relation between the schedule polyhedron and the unit length schedules.

Lemma 1. *(1) Each point in the schedule polyhedron P is a feasible link time vector $\boldsymbol{t}$, and (2) each feasible link time vector $\boldsymbol{t}$ is a point in P.*

Proof. The proof uses the Edmonds' matching polyhedron theorem [14]. A feasible schedule $\mathcal{S}$ consists of $N \geq 1$ slots. Each slot contains a set of links from G that is a matching and therefore corresponds to a vertex of the matching polyhedron Q. Since Q has the same formulation as P, except the variables are binary, P and Q has the same set of vertices. Since $\mathcal{S}$ has a length of 1, the link time vector $\boldsymbol{t}$ of $\mathcal{S}$ is a convex combination of the vertices of P. So it is a point in P. On the other hand, a point in P can be written as a convex combination of all vertices of P where each vertex corresponds to a slot.
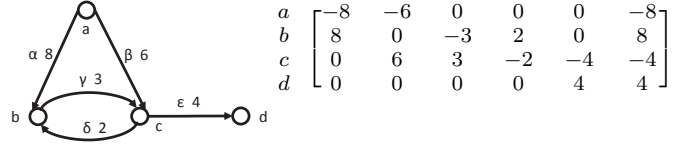
Therefore, each point in P corresponds to a feasible unit time schedule. See the details in Appendix A. $\square$

V. MAXIMUM THROUGHPUT FAIR SCHEDULING

Having established the relation between a schedule and matchings, we now investigate the problem of *maximum throughput fair scheduling* (MTFS) for backhaul networks. The goal of the problem is to maximize the downlink *network throughput* under the condition that the *max-min fairness* [15], [16] in throughput is achieved at the mmBSs. The MTFS problem serves as one example of our method for scheduling optimization in mmWave backhaul networks.

Definition 2 (Maximum Throughput Fair Schedule). *Given a backhaul network G and a unit time schedule $\mathcal{S}$, let the throughput vector of $\mathcal{S}$ be $\boldsymbol{h}^{\mathcal{S}} = [h_v^{\mathcal{S}} | v \in \mathcal{W}]$, where $h_v^{\mathcal{S}}$ denotes the downlink throughput of an mmBS node v.*
(i) A feasible unit time schedule $\mathcal{S}_f$ is said to satisfy the max-min fairness criteria if $\min_{v \in \mathcal{W}} h_v^{\mathcal{S}_f} \geq \min_{v \in \mathcal{W}} h_v^{\mathcal{S}}$ for any feasible unit time schedule $\mathcal{S}$. Such $\min_{v \in \mathcal{W}} h_v^{\mathcal{S}_f}$ is called the max-min throughput.
(ii) A feasible unit time schedule $\mathcal{S}^*$ is a solution of the MTFS problem if $\mathcal{S}^*$ has achieved the maximum network throughput $\sum_{v \in \mathcal{W}} h_v^{\mathcal{S}_f}$ among all possible feasible unit time schedule $\mathcal{S}_f$ satisfying the max-min fairness criteria in (i).

In the following, we present our general optimization method—*schedule oriented optimization*.

A. Schedule oriented optimization

The schedule oriented optimization solves a linear optimization problem, the solution to which is directly the optimal schedule. For the mathematical formulation of the optimization problem, we construct the *node-matching matrix*.

Definition 3 (Node-Matching Matrix). *Given a directed graph $G = (\mathcal{V}, \mathcal{E})$. Suppose the number of all possible matchings of G is K. Then the node-matching matrix $\mathsf{A} = [a_{i,j}]$ is a $|\mathcal{V}| \times K$ matrix, whose elements is defined as follows: $a_{i,j} = c$, if there is a link with capacity c entering node i in the j-th matching; $a_{i,j} = -c$, if there is a link with capacity c leaving node i in the j-th matching. Otherwise, $a_{i,j} = 0$.*

As we will see, the node-matching matrix helps in formulating the throughput constraints at individual nodes. Fig. 4 gives an example of node-matching matrix for a graph.

Let A be the node-matching matrix of the backhaul network G, we define $\mathsf{A}^{\mathcal{W}}$ as the submatrix of A, which consists only of the rows of A related to the nodes in $\mathcal{W}$ (mmBSs). As

we pointed out in Sec. IV, the link set scheduled in each slot of a schedule must be a matching in G. We define $\boldsymbol{t}^{\mathcal{S}}$ as a $K \times 1$ *slot length vector*, each element of which is the length of a potential slot. Let the minimum throughput among all mmBSs be θ. Then we can solve the MTFS problem in two steps: (i) maximizing θ (such θ is the max-min throughput) and (ii) computing the optimal schedule $\mathcal{S}^*$ that offers the highest network throughput, given the max-min throughput θ.

Linear programs for the MTFS problem. The linear program to maximize θ for step (i) of the MTFS problem can be formulated as follows,

$$\text{maximize} \quad \theta \tag{2a}$$
$$\text{subject to} \quad \mathsf{A}^{\mathcal{W}} \boldsymbol{t}^{\mathcal{S}} \geq \theta \mathbf{1} \tag{2b}$$
$$\mathbf{1}^{\mathsf{T}} \boldsymbol{t}^{\mathcal{S}} = 1 \text{ and } \boldsymbol{t}^{\mathcal{S}} \geq \mathbf{0}, \tag{2c}$$

where $\mathbf{1}$ and $\mathbf{0}$ represent the all-one and all-zero column vector, respectively. The superscript "T" denotes the vector transposition. (2b) is the constraint that the throughput at each mmBS should be at least θ. (2c) is the constraint that the schedule length should be unit time. The feasibility of the schedule is implicitly guaranteed by the formulation in terms of all possible matchings.

After we have computed θ from (2), we can formulate the linear program that maximizes the *network throughput*, i.e., the total throughput of all mmBSs under the condition that each mmBS has throughput at least θ.

$$\text{maximize} \quad \boldsymbol{c}^{\mathsf{T}} \boldsymbol{t}^{\mathcal{S}} \tag{3a}$$
$$\text{subject to} \quad \text{(2b), and (2c).} \tag{3b}$$

Here, $\boldsymbol{c}$ is the *capacity vector* whose element c_j is the cumulative capacity of all eNB-to-mmBS links in the j-th matching M_j, i.e., $c_j = \sum_{(v,v') \in M_j, v \in \mathcal{R}} c_{(v,v')}$, where (v, v') denotes the link from node v to node v'. Note that θ is a variable in (2), but it is a constant in (3).

The apparent difficulty in solving (2) and (3) is the huge number of elements in $\boldsymbol{t}^{\mathcal{S}}$ (same as the number of matchings in G, which is exponential to the number of vertices in G). Yet, we will show that we can still solve it in polynomial time.

Theorem 1. *The MTFS problem can be solved in polynomial time with the ellipsoid algorithm [17].*

Proof. This proof uses a similar technique to the proof of Theorem 2 in [18], which states that the fractional edge coloring can be solved in polynomial time by the ellipsoid algorithm. See the details in Appendix B. □

Although polynomial, in practice the ellipsoid algorithm almost always takes longer than the simplex algorithm. In the following, we propose algorithms based on the revised simplex algorithm [19] which does not require the generation of all columns of $\mathsf{A}^{\mathcal{W}}$. Conceptually, the algorithms first create a feasible schedule. Then in each iteration, to improve the optimization objective, we replace one slot in the schedule by another matching (a set of simultaneous links) while keeping the schedule feasible, until the optimum is reached. The maximum weighted matching algorithm [14] is used to choose the matching (column) to enter the basis (schedule).

Algorithm 1: Compute the max-min throughput θ

1 Set the basis $\mathsf{B} = \mathsf{B}_0$ corresponding to the initial schedule $\mathcal{S}_0$;
2 while *True* **do**
3   Compute the dual variable $\boldsymbol{p}^{\mathsf{T}} = \boldsymbol{f}_{\mathsf{B}}^{\mathsf{T}} \mathsf{B}^{-1}$;
4   Set weight $w_{(v_i,v_j)}$ to each link (v_i, v_j) of G as follows.

$$w_{(v_i,v_j)} = \begin{cases} c_{(v_i,v_j)}(p_j - p_i) & \text{if } v_i, v_j \in \mathcal{W} \\ c_{(v_i,v_j)} p_j & \text{otherwise } v_i \in \mathcal{R}, v_j \in \mathcal{W} \end{cases}$$

    Do max weighted matching on G. Let the optimal matching be M, compute $\eta_1 = -\sum_{e \in M} w_e - p_{W+1}$;
5   Compute $\eta_2 = -1 + \sum_{k=1}^{W} p_k$;
6   Compute $\eta_3 = \min_{1 \leq k \leq W} p_k$;
7   Compute $\eta = \min(\eta_1, \eta_2, \eta_3)$ and let the corresponding column be $\boldsymbol{u}_\eta \in \mathsf{U}$;
8   **if** $\eta \geq 0$ **then**
9   **return** the optimal θ and $\mathsf{B}_\theta = \mathsf{B}$;
10   **else**
11   Update B by replacing a column of B with $\boldsymbol{u}_\eta$ according to the simplex algorithm;
12   **end**
13 end

B. Solving the MTFS problem

To optimize θ, we need an initial basic feasible solution to (2). Suppose that the backhaul network G is connected, otherwise there are mmBSs unreachable from the eNB. We perform a *breath-first-search* (BFS) starting from an arbitrary expanded eNB, say eNB_1. The result is a tree T that spans eNB_1 and all mmBSs. T has exactly W edges. The initial schedule $\mathcal{S}_0$ is constructed as follows: $\mathcal{S}_0$ has W slots, each of which contains one link in T. Moreover, it is required that the throughputs of all mmBSs are the same and the schedule takes exactly unit time. It is obvious that the initial solution is unique. We convert the linear program (2) to the standard form (4) by introducing W *surplus variables* s_i as follows.

$$\text{minimize} \quad \boldsymbol{f}^{\mathsf{T}} \boldsymbol{x} \tag{4a}$$
$$\text{subject to} \quad \mathsf{U} \boldsymbol{x} = \boldsymbol{g} \text{ and } \boldsymbol{x} \geq \mathbf{0}, \tag{4b}$$

where $\mathsf{U} \triangleq [\mathsf{U}^1 | \mathsf{U}^2 | \mathsf{U}^3] \triangleq \begin{bmatrix} \mathsf{A}^{\mathcal{W}} & -\mathbf{1} & -\mathsf{I} \\ \mathbf{1}^{\mathsf{T}} & 0 & \mathbf{0}^{\mathsf{T}} \end{bmatrix}$, $\boldsymbol{f}^{\mathsf{T}} = [\mathbf{0}^{\mathsf{T}} | -1 | \mathbf{0}^{\mathsf{T}}]$, $\boldsymbol{x}^{\mathsf{T}} \triangleq [(\boldsymbol{t}^{\mathcal{S}})^{\mathsf{T}} | \theta | \boldsymbol{s}^{\mathsf{T}}]$, and $\boldsymbol{g}^{\mathsf{T}} \triangleq [\mathbf{0}^{\mathsf{T}} | 1]$. Alg. 1 shows the computation of the max-min throughput θ.

The basis B is a square matrix that consists of $W + 1$ columns from U. $\boldsymbol{f}_{\mathsf{B}}$ is the elements of $\boldsymbol{f}$ corresponding to the basis B. The lines 4, 5 and 6 compute the minimum reduced cost of a column in the matrices $\mathsf{U}^1, \mathsf{U}^2$ and U^3 respectively. To decrease $-\theta$, we need to find a column of U, $\boldsymbol{u}_k$ that has negative reduced cost $f_k - \boldsymbol{p}^{\mathsf{T}} \boldsymbol{u}_k < 0$ to enter the basis. In the algorithm, we find the column $\boldsymbol{u}_\eta$ in U that produces the minimum reduced cost η. If $\eta \geq 0$, then no columns can be used to decrease $-\theta$, thus we have reached the optimum.

Let the final basis in computing θ be B_θ. To directly use B_θ as the initial basis to the solution of step (ii) of the MTFS problem, we add an artificial scalar variable $y \geq 0$ to (3) and replace the constraint $\mathsf{A}^{\mathcal{W}} \boldsymbol{t}^{\mathcal{S}} \geq \theta \mathbf{1}$ with $\mathsf{A}^{\mathcal{W}} \boldsymbol{t}^{\mathcal{S}} - \mathbf{1}y \geq \theta \mathbf{1}$. Since θ is the max-min throughput, the feasible y must be 0. Hence, the optimal solution (maximum network throughput) to

Algorithm 2: Solving the MTFS problem

1 Set the basis $\mathsf{B} = \mathsf{B}_\theta$;
2 **while** *True* **do**
3   Compute the dual variable $\boldsymbol{p}^\mathsf{T} = \boldsymbol{f}_\mathsf{B}^\mathsf{T}\mathsf{B}^{-1}$;
4   Set weight $w_{(v_i,v_j)}$ to each link (v_i, v_j) of G as follows.

$$w_{(v_i,v_j)} = \begin{cases} c_{(v_i,v_j)}(p_j - p_i) & \text{if } v_i, v_j \in \mathcal{W} \\ c_{(v_i,v_j)}(p_j + 1) & \text{otherwise } v_i \in \mathcal{R}, v_j \in \mathcal{W} \end{cases}$$

    Do max weighted matching on G. Let the optimal matching be M, compute $\eta_1 = -\sum_{e \in M} w_e - p_{W+1}$;
5     $\eta_2 = \sum_{k=1}^{W} p_k$;
6     $\eta_3 = \min_{1 \le k \le \mathcal{W}} p_k$;
7   Compute $\eta = \min(\eta_1, \eta_2, \eta_3)$ and let the corresponding column be $\boldsymbol{u}_\eta \in \mathsf{U}$;
8   **if** $\eta \ge 0$ **then**
9   **return** the optimal schedule $\mathcal{S}^*$ corresponding to B;
10   **else**
11   Update B by replacing a column of B with $\boldsymbol{u}_\eta$;
12   **end**
13 **end**

(3) is unaffected. Again, we convert (3) into the standard form of (4), which is solvable with the revised simplex algorithm.

In the standard form, U remains unchanged, we redefine $\boldsymbol{f}^\mathsf{T} \triangleq \begin{bmatrix} -\boldsymbol{c}^\mathsf{T} \,|\, 0 \,|\, \boldsymbol{0}^\mathsf{T} \end{bmatrix}$, $\boldsymbol{x}^\mathsf{T} \triangleq \begin{bmatrix} (\boldsymbol{t}^\mathcal{S})^\mathsf{T} \,|\, y \,|\, \boldsymbol{s}^\mathsf{T} \end{bmatrix}$, and $\boldsymbol{g}^\mathsf{T} \triangleq \begin{bmatrix} \theta\boldsymbol{1}^\mathsf{T} \,|\, 1 \end{bmatrix}$. The optimization algorithm is similar to Alg. 1 and is outlined in Alg. 2. Since the basis B is a square matrix of $W + 1$ dimension, it follows that the optimal schedule $\mathcal{S}^*$ contains no more than $W + 1$ slots. Additionally, since the links on a flow from the eNB to a destination mmBS may not be scheduled in sequential order, some transmission opportunities of the flow in the first few frames may be wasted. Therefore, maximum throughput is achieved in the long-term.

C. Generalization

The scheduled-oriented optimization method illustrated by the optimal MTFS algorithm is quite general. It can solve any scheduling problem that can be formulated as a linear program whose variables are link times and QoS metrics. For example, it can optimize for the constraint that each mmBS has a minimum throughput requirement. Another example is that the proposed method can optimize the energy consumption as it can be translated into the minimization of total transmission time in a schedule. We do not further elaborate on them due to the space limitation. Moreover, in Sec. VII, we extend the optimization method to backhaul and access networks, as well as to multi-RF chains at each node.

VI. Edge-coloring based approximation algorithm

In Sec. V, we proposed an optimal joint routing and scheduling algorithm for mmWave backhaul networks. Although it is optimal, it may have a high runtime (c.f. the evaluation in Sec. VIII) when the number of mmBS nodes is large. Hence, we propose a run-time efficient *edge-coloring (EC) based approximation algorithm* that has a provable performance bound. The EC algorithm follows a two-step approach of (i) computing the link time and (ii) scheduling within unit time.

A. Step (i): computing link time

To precisely compute the link time, we need to include all the constraints of the schedule polyhedron (1). Since the number of odd set constraints (1b) is huge, which leads to a high runtime for the optimization, instead we use a small set of constraints that is a necessary but not sufficient condition for a feasible unit time schedule. The selection of the new set of constraints is based on the following observation.

Let $G = (\mathcal{V}, \mathcal{E})$ be the backhaul network before the eNB expansion (Sec. IV-A) and $G^{\mathcal{W}}$ be the subgraph of G, which contains only the mmBSs $\mathcal{W}$ and the links among them. We define $\nu = \lfloor \frac{W}{2} \rfloor$ as an upper bound of the maximum number of mmBS-to-mmBS links that can be active simultaneously[1]. We assume the number of RF chains at the eNB satisfies $R \le L$, where L is the number of mmBSs that are directly connected to the eNB, because L RF chains is enough to serve the mmBSs. We have the following observation.

Observation 1. *If k mmBS-to-mmBS links are active at a time t, then at most $\min(R, W - 2k)$ eNB-to-mmBS links can be active at t, each using one RF chain of the eNB. Hence, at least $R - \min(R, W - 2k) = \max(0, R - W + 2k)$ RF chains of the eNB are idle at t.*

For a schedule of unit time, we define t'_k as the time in which exactly k mmBS-to-mmBS links are active. Each feasible schedule should be subject to the following constraints.

$$\sum_{k=1}^{\nu} t'_k \le 1, \text{ and } t'_k \ge 0 \quad \forall k \in \{1, 2, ..., \nu\} \tag{5a}$$

$$\sum_{k=1}^{\nu} k \cdot t'_k = \sum_{e \in \{(v,v') \in \mathcal{E} \,:\, v, v' \in \mathcal{W}\}} t_e \tag{5b}$$

$$\sum_{e \in \delta(\text{eNB})} t_e \le R - \sum_{k=1}^{\nu} \max(0, R - W + 2k)t'_k \tag{5c}$$

$$\sum_{e \in \delta(v)} t_e \le 1 \quad \forall v \ne \text{eNB}, \text{ and } t_e \ge 0 \quad \forall e \in \mathcal{E}. \tag{5d}$$

(5b) formulates the total mmBS-to-mmBS transmission time in terms of the variables t'_k and t_e (link time of e), respectively. (5c) shows that the total eNB-to-mmBS transmission time should be no more than R minus the minimum idle time of the RF chains at the eNB. (5d) expresses the single RF chain constraint on mmBSs.

We substitute the precise constraint set to link times (1) with the constraints in (5). The advantage is the low runtime and small memory complexity of linear programming due to the following reason. The total number of constraints in (1) and (5) are $O(2^{W+R})$ (exponential) and $O(W^2)$ (polynomial), respectively. Moreover, with the polynomial number of constraints, the computation of link time can be carried out by off-the-shelf linear optimization tools. However, the link time vector that satisfies (5) may be infeasible in unit time, because satisfying the constraints in (5) is a necessary but not sufficient condition for a feasible unit time schedule.

---

[1]The active links form a matching in $G^{\mathcal{W}}$. According to the definition of matching, the number of active links is upper-bounded by $\lfloor \frac{W}{2} \rfloor$.

Algorithm 3: EC-based scheduling.

1 **Reduce graph**. Given the link time vector $\boldsymbol{t}$, we remove edges in G with zero link time and call the subgraph G_r;
2 **Expand eNB**. We perform the eNB expansion on G_r. Let the expanded eNBs be $\text{eNB}_1, ..., \text{eNB}_R$. For a given link (eNB, v) in G_r with link time $t_{(\text{eNB},v)}$, we set the link time of the links $(\text{eNB}_k, v), k = 1, ..., R$ to $\frac{t_{(\text{eNB},v)}}{R}$. We call the graph after eNB expansion $G_v = (\mathcal{V}_v, \mathcal{E}_v)$;
3 **Create multigraph and assign link time**. We create the coloring graph $G_m = (\mathcal{V}_m, \mathcal{E}_m)$, which has the same vertex set as $\mathcal{V}_v$, and its edges is defined as follows. For each $e \in \mathcal{E}_v$ between two nodes v and v' with link time t_e, we install $\left\lceil \frac{t_e}{t^g} \right\rceil$ edges between v and v' in G_m. Among these edges, $\left\lceil \frac{t_e}{t^g} \right\rceil - 1$ edges are assigned t^g link time, and the left edge is assigned $\text{mod}(t_e, t^g)$ link time (mod is the modulo operation);
4 **Coloring and scheduling**. We perform edge coloring on G_m. Suppose that G_m can be edge-colored with κ colors. For those edges colored by the i-th color, $i = 1, \ldots, \kappa$, we schedule the corresponding links in the i-th slot (a slot has the length t^g);
5 **Scale**. The schedule is now of length κt^g. If $\kappa t^g > 1$, we scale the total time length with the factor $\frac{1}{\kappa t^g}$;

Specifically, for the MTFS problem, the linear program for computing the max-min throughput θ is

$$\text{maximize} \qquad \theta \qquad \qquad (6a)$$
$$\text{subject to} \quad \sum_{e \in \delta^-(v)} c_e t_e - \sum_{e \in \delta^+(v)} c_e t_e \geq \theta \quad \forall v \in \mathcal{W} \quad (6b)$$
$$\text{and } (5),$$

where $\delta^-(v)$ and $\delta^+(v)$ are the set of links coming into node v and the set of links leaving v, respectively.

With the optimal θ, we compute the link time for the MTFS.

$$\text{maximize} \quad \sum_{e \in \delta^+(\text{eNB})} c_e t_e$$
$$\text{subject to} \quad \text{constraints in } (6). \qquad (7)$$

B. Step (ii): edge-coloring based scheduling

After we obtain the link time, the next step is to generate a unit time schedule. The approximation algorithm is based on the idea of edge-coloring of multigraphs (graphs allowing multiple edges between two nodes). A proper edge-coloring assigns a color to each edge in a graph such that any two adjacent edges (sharing one or two common nodes) are assigned different colors. Obviously, the set of edges $\mathcal{E}_\lambda$ of a color λ must be a matching. Hence, $\mathcal{E}_\lambda$ corresponds to a slot and an edge coloring scheme corresponds to a schedule. The EC-based scheduling takes a parameter *granularity* $t^g \in (0, 1]$, which is the quantization of the link time. A smaller t^g typically leads to better schedules at the cost of longer runtime. Alg. 3 shows the process of the EC-based scheduling.

C. Performance analysis of the EC-based scheduling

The following lemma shows that Alg. 3 (step (ii) of the EC algorithm) reduces the performance metric μ [2] and link times t_e by a factor of $\frac{1}{\kappa t^g}$, if the $\kappa t^g > 1$. Therefore, a high quality

---

[2]μ is the optimization goal such as throughput, energy consumption, etc.

edge-coloring heuristic (small κ) [20] and a small t^g improve the schedule performance.

Lemma 2. *Suppose that after step (i) of the EC algorithm, each link e of G_v has link time t_e, and the performance metric is $\mu \geq 0$. Moreover, assume that if a schedule is scaled by $\rho > 0$, then μ is also scaled by ρ. Therefore, after step (ii), the final link time is $t'_e = \min(\frac{t_e}{\kappa t^g}, t_e)$ and the final performance metric is $\mu' = \min(\frac{\mu}{\kappa t^g}, \mu)$.*

Proof. If $\kappa t^g \leq 1$, then G_v can be scheduled in unit time, and $t'_e = t_e$ and $\mu' = \mu$. On the other hand, if $\kappa t^g > 1$, then G_v needs κt^g time to schedule. To fit in the unit time schedule, we perform the scaling. Afterwards, $t'_e = \frac{t_e}{\kappa t^g}$ and $\mu' = \frac{\mu}{\kappa t^g}$. $\qquad\square$

Since minimum edge coloring of an arbitrary graph is NP-complete [21], we have to employ approximation algorithms. We choose a simple multigraph edge-coloring algorithm by Karloff et al [22]. It uses at most $3\lceil\Delta(G)/2\rceil$ colors, where $\Delta(G)$ is the maximal node degree of a multigraph G. The following lemma gives the upper bounds on $\Delta(G_m)$, the number of vertices $|\mathcal{V}_m|$ and edges $|\mathcal{E}_m|$ of G_m.

Lemma 3. $\Delta(G_m) \leq W + R + 1/t^g - 1$. $|\mathcal{V}_m| \leq W + R$ and $|\mathcal{E}_m| < \frac{1}{2}\left(W^2 + (2R-1)W + \frac{W+R}{t^g}\right)$.

Proof. G_v contains $W + R$ nodes due to the eNB expansion. For a expanded eNB node, the maximum degree is no more than W. For a mmBS, the maximum degree is no more than $W - 1 + R$ since the directed graph G_v contains no cycles as all cycles can be eliminated by shortening the link time. So $\Delta(G_v) \leq W + R - 1$. G_m is transformed from G_v by installing $\left\lceil \frac{t_e}{t^g} \right\rceil$ edges for each edge e in G_v. Therefore, the degree of a node v in G_m is

$$\deg(v) = \sum_{\substack{e \in \delta(v) \\ v \in \mathcal{V}_v}} \left\lceil \frac{t_e}{t^g} \right\rceil < \sum_{\substack{e \in \delta(v) \\ v \in \mathcal{V}_v}} \left(\frac{t_e}{t^g} + 1 \right) \leq \frac{1}{t^g} + W + R - 1$$

In addition, we have $|\mathcal{V}_m| = |\mathcal{V}_v| \leq W + R$. Using the *degree sum formula*, the number of edges in G_m is

$$|\mathcal{E}_m| = \frac{1}{2} \sum_{v \in \mathcal{V}_m} \deg(v) < \frac{1}{2} \sum_{v \in \mathcal{V}_v} \sum_{e \in \delta(v)} \left(\frac{t_e}{t^g} + 1 \right)$$
$$\leq \frac{1}{2}\left(W^2 + (2R-1)W + \frac{W+R}{t^g} \right). \qquad\square$$

Since $R \leq W$ (W RF chains is sufficient to serve all mmBSs), from the above Lemma, we have $\Delta(G_m) = O(W + \frac{1}{t^g})$. $|\mathcal{V}_m| = O(W)$ and $|\mathcal{E}_m| = O(W^2 + \frac{W}{t^g})$.

In the following, we show the quality and time complexity of the step (ii) of the EC algorithm. Practically, step (i) is always much faster than step (ii).

Theorem 2. *Let the performance metric after step (i) be $\mu \geq 0$. Then step (ii) achieves the performance metric $\mu' > \frac{2}{3[(W+R+1)t^g+1]}\mu$ and it has time complexity of $O\left(\left[W^2 + \frac{W}{t^g}\right]\log(W + \frac{1}{t^g})\right)$.*

Proof. The Karloff's algorithm uses $\kappa \leq 3\lceil\Delta(G_m)/2\rceil$ colors. Due to Lemma 3, $\kappa \leq 3\lceil(W + R + \frac{1}{t^g} - 1)/2\rceil$. Hence,

$$\kappa t^g < 3t^g\left(\frac{W + R + \frac{1}{t^g} - 1}{2} + 1\right) = \frac{3}{2}(W + R + 1)t^g + \frac{3}{2}.$$

According to Lemma 2, the final performance metric

$$\mu' = \min(\frac{1}{\kappa t^g}, 1)\mu > \frac{2}{3[(W + R + 1)t^g + 1]}\mu$$

The time complexity of step (ii) is determined by Karloff's edge coloring algorithm, which has the same time complexity as the perfect edge coloring of a bipartite graph of maximum degree $\lceil\Delta(G_m)/2\rceil$ and number of edges $O(|\mathcal{E}_m|)$. Since the time complexity of the perfect edge coloring of a graph with $|\mathcal{E}|$ edges and degree Δ is $O(|\mathcal{E}|\log\Delta)$ [23], the time complexity of step (ii) is $O(|\mathcal{E}_m|\log(\lceil\Delta(G_m)/2\rceil)) = O(\lceil\frac{W}{t^g} + W^2\rceil\log(W + \frac{1}{t^g}))$. □

For the MTFS problem, let the optimal max-min throughput be θ^*. Since step (i) of the EC algorithm uses a looser constraint set than the precise set of the schedule polyhedron, it gives a $\theta \geq \theta^*$. From Theorem 2, we have that the final max-min throughput of the EC algorithm $\theta' > \frac{2}{3[(W+R+1)t^g+1]}\theta^*$.

In a typical backhaul network, we have $W \gg R$, so $\frac{2}{3[(W+R+1)t^g+1]} \approx \frac{2}{3(Wt^g+1)}$. This means to keep a constant performance quality, we can choose t^g to be inversely proportional to W, i.e., a bigger network requires a smaller t^g. If t^g is so selected, then the time complexity is $O(W^2\log(W))$, which is quite scalable with the number of mmBSs, and thus feasible at the eNB in practice. Moreover, by setting the granularity $t^g \to 0$, the performance metric approaches $\mu' > \frac{2}{3}\mu$. For the MTFS problem, this means $\theta' > \frac{2}{3}\theta^*$.

VII. EXTENSION TO MORE GENERAL SCENARIOS

In this section, we show that our schedule-oriented optimization method proposed in Sec. V and the edge-coloring based approximation algorithm in Sec. VI can be extended to more general scenarios of (i) backhaul and access networks and (ii) multiple RF chains at each node.

A. Extension to backhaul and access networks

The backhaul and access networks add an additional layer of UEs to the backhaul networks. Each UE has a single RF chain and is allowed to have links with one or more mmBSs. Let $\mathcal{U}$ denote the set of UEs. For the downlink traffic, only one-directional links from mmBSs to UEs exist.

We illustrate as an example the solution of the MTFS problem. The max-min fairness in throughput is now defined for the UEs, as they are the destinations. Let G be the backhaul and access network after eNB expansion, and A be the node-matching matrix of G. We define $A^{\mathcal{W}}$ and $A^{\mathcal{U}}$ as the submatrices of A related to the nodes in $\mathcal{W}$ (mmBSs) and in $\mathcal{U}$ (UEs), respectively. The linear program (2) to compute max-min throughput θ needs to be modified as follows: (2b) should be replaced by the constraints of (8).

$$A^{\mathcal{U}}\boldsymbol{t}^{\mathcal{S}} \geq \theta\boldsymbol{1} \text{ and } A^{\mathcal{M}}\boldsymbol{t}^{\mathcal{S}} = \boldsymbol{0} \qquad (8)$$
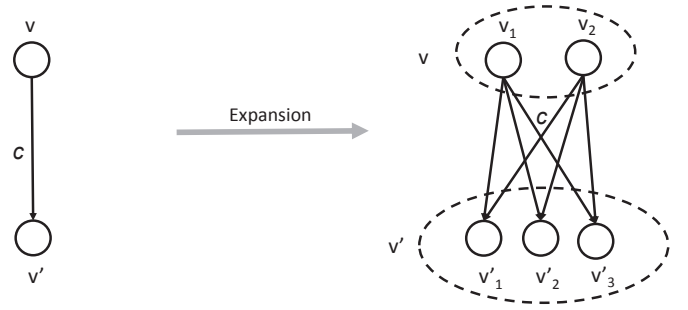

Fig. 5. Node expansion. v has 2 RF chains and v' has 3 RF chains.

here, (8) expresses the constraint that the throughput at a UE must be at least θ and each mmBS is a pure relay. Obviously, the new MTFS problem can be solved with the same optimization technique as proposed in Sec. V.

As for the EC algorithm, we need to replace the data flow constraint of (6b) with the following (9).

$$\sum_{e\in\delta^-(v)} c_e t_e - \sum_{e\in\delta^+(v)} c_e t_e = 0 \quad \forall v \in \mathcal{W} \qquad (9a)$$

$$\sum_{e\in\delta^-(v)} c_e t_e - \sum_{e\in\delta^+(v)} c_e t_e \geq \theta \quad \forall v \in \mathcal{U} \qquad (9b)$$

B. Extension to multiple RF chains at each node

Now we remove the restriction that all nodes except the eNB have single RF chain. The technique to deal with this problem is the so-called *node expansion* which extends the eNB expansion.

Node expansion. Let G be a directed graph representing the network. We create a expanded graph G' as follows: for each node v in G, we create R_v *expanded nodes* $v_1, ..., v_{R_v}$ in G' where R_v is the number of RF chains at v. The expanded nodes of v are collectively called a *super node* v in G'. Moreover, if there is a link of capacity c between node v and node v' in G, then we install $R_v \cdot R_{v'}$ links between all combinations of $v_i, i = 1, ..., R_v$ and $v'_j, j = 1, ..., R_{v'}$. Each link (v_i, v'_j) is assigned the capacity c. An example is shown in Fig. 5.

After the node expansion, the constraint of the RF chains is implicitly guaranteed by the matchings in G'. For the MTFS problem, the definition of the node-matching matrix $A^{\mathcal{W}}$ needs adaptation because now multiple links can be incident to a super node in a matching in G'. A row of $A^{\mathcal{W}}$ corresponds to an mmBS super node (a collection of expanded nodes) and a column of $A^{\mathcal{W}}$ corresponds to a matching in G'. Therefore, an element of the matrix $a_{i,j}^{\mathcal{W}}$ has the value of the sum capacity of all links entering the i-th mmBS super node in the j-th matching minus the sum capacity of all links leaving the i-th mmBS super node in the j-th matching.

As for the EC algorithm, we need to modify the necessary schedule constraints in (5). Since the maximum number of simultaneous mmBS-to-mmBS links increases for R times when each mmBS has R RF chains. This leads to R times more variables of t'_k in (5), which may lead to long computation time in linear program. So, we delete the constraints on t'_k.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Distance between 2 mmBSs, d_g	100 m
Path loss parameters α, β, σ in $PL(d) = \alpha + 10\beta \log_{10} d + \xi$	LOS: $\alpha = 61.4, \beta = 2, \sigma = 5.8$ NLOS: $\alpha = 72, \beta = 2.92, \sigma = 8.7$
Transmission power, p_{tx}	30 dB
Directivity gain, g_{x}	30 dB
Bandwidth, b	1 GHz
Noise $N_0 = kT_0 + F + 10\log_{10} b$	$kT_0 = -174$ dBm/Hz, $F = 4$ dB
Minimum SINR threshold, τ	-5 dB

The following simple set of necessary schedule constraints is used to replace (5),

$$\sum_{e \in \delta(v)} t_e \leq R_v \quad \forall v \in \mathcal{V}, \text{ and } t_e \geq 0 \quad \forall e \in \mathcal{E}.$$

Another modification is to replace the *expand eNB* step in Alg. 3 with the following step.

Expand node. We perform the node expansion on G_r. Let the resulting graph be G_v. For a given link (v, v') in G_r with link time $t_{(v,v')}$, we assign the link time $\frac{t_{(v,v')}}{R_v R_{v'}}$ to the links (v_i, v'_j) in G_v for all combinations of $i = 1, ..., R_v$ and $j = 1, ..., R_{v'}$.

VIII. NUMERICAL EVALUATION

In this section, we evaluate the optimal MTFS algorithm and EC-based approximation algorithm in terms of max-min throughput, network throughput and runtime efficiency.

A. Evaluation setting

We simulate a mmWave backhaul network, where $n \times n$ mmBSs are placed on the intersections of a $n \times n$ grid and the eNB is placed in the center of the grid. The distance between two neighboring mmBSs is d_g. The capacity of each link is calculated with the channel model described in Sec. III. We assume a carrier frequency of 28 GHz. The channel state between any two nodes is simulated according to the statistical model derived from the real-world measurement [24]. The channel state has three possibilities—LOS (line-of sight), NLOS (non line-of-sight) or outage. The simulation parameters are listed in Tab. I.

The proposed algorithms are implemented in MATLAB, with the exception that the optimal MTFS algorithm uses a C++ implementation for maximum weighted matching [25].

B. mmBS with single RF chain

We evaluate the MTFS scheduling by varying the number of mmBSs from 4×4 to 16×16. The eNB has $R = 10$ RF chains and all the other nodes have single RF chain. For each network size, 30 instances of link capacities are randomly generated and then the network is scheduled for the MTFS problem. The performance results are shown in Fig. 6. As expected, the optimal MTFS algorithm (OPT-MTFS) always attains the highest max-min throughput (Fig. 6(a)). In contrast, the max-min throughput of the EC-based approximation algorithm (EC) is smaller. However, the value is significantly better

than the theoretical lower bound of Theorem 2 (dashed lines in the figure). It goes up with an increase in granularity (corresponding to a lower t^g). Practically, t^g in the range of 0.01 to 0.001 is ideal for the network size of up to 200 nodes, as on average, the EC algorithm achieves 70% to 90% of the optimal max-min throughput. The price for the high max-min throughput is a decrease in runtime efficiency. For a network of 256 mmBSs, the EC algorithm with $t^g = 0.001$ runs 100x faster than the optimal MTFS algorithm (Fig. 6(b), note the log scale). Moreover, with the increase of the number of nodes, the runtime of the EC algorithm grows more slowly than the optimal MTFS algorithm, which shows the better scalability of the former for large networks.

As the goal of the MTFS is to maximize the network throughput under the fairness condition, we also compare the network throughput of the optimal MTFS algorithm, the EC algorithm and the unconditional maximum network through-put (MAX-TPUT). The MAX-TPUT achieves the maximum network throughput for a given network. It is obtained when the $\min(R, L)$ (L is the number of mmBSs directly connected to the eNB) eNB-to-mmBS links with the highest capacities are active throughout the unit schedule and all the other links are inactive. If the number of mmBSs $W > L$, some of the mmBSs will have zero throughput. This is the worst case with respect to max-min fairness in throughput. Our evaluation results show that, on average, max-min fairness limits the network throughput to be approximately half of the maximum value. Since the simulated backhaul network is well-connected, the network throughput is in most cases equal to the max-min throughput times the number of mmBSs. Therefore, the relative performance of the network throughput between the optimal MTFS algorithm and the EC algorithm at different granularities is almost the same as that of the max-min throughput in Fig. 6(a).

C. mmBS with multiple RF chains

We now evaluate the performance for the situation that each mmBS is equipped with multiple RF chains. For that purpose, we simulate a backhaul network with 10×10 mmBSs and evaluate the cases that the eNB has 10 RF chains and each mmBS has $R_{\mathcal{W}}$ RF chains, with $R_{\mathcal{W}}$ varying from 1 to 10. For each given $R_{\mathcal{W}}$, 30 instances of random link capacities are generated.

As shown in Fig. 7(a), the max-min throughput goes up steadily with $R_{\mathcal{W}}$. The optimal max-min throughput at $R_{\mathcal{W}} = 10$ is over 4 times higher than the value at $R_{\mathcal{W}} = 1$. The difference in performance is due to the larger number of simultaneous links in the setting of multi-RF-chain mmBSs. Evaluation results show that the number of simultaneous links is almost proportional to $R_{\mathcal{W}}$, as node expansion has increased the number of nodes for $R_{\mathcal{W}}$ times. Therefore, to attain higher throughput at each mmBS, an option is to equip mmBSs with multiple RF chains. However, the run time of the optimal MTFS algorithm also increases with $R_{\mathcal{W}}$ (Fig. 7(b)). The extra time is spent in the maximum weighted matching in an expanded network with roughly $R_{\mathcal{W}}$ times more nodes and $R_{\mathcal{W}}^2$ times more edges. By using the EC-algorithm with
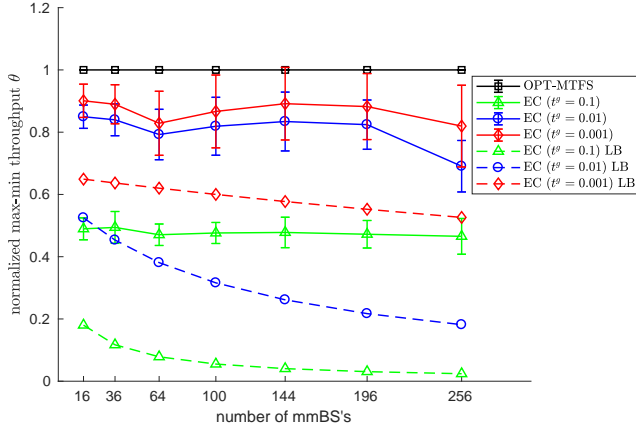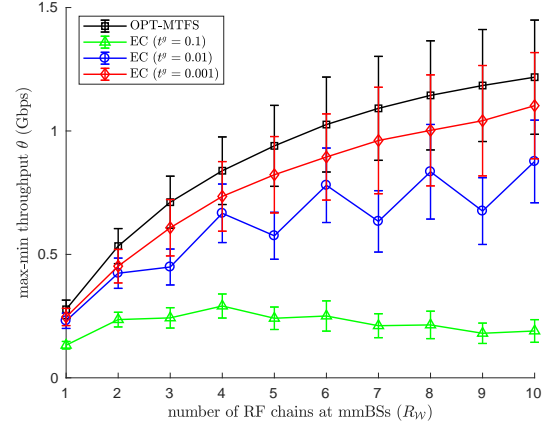
(a) max-min throughput θ
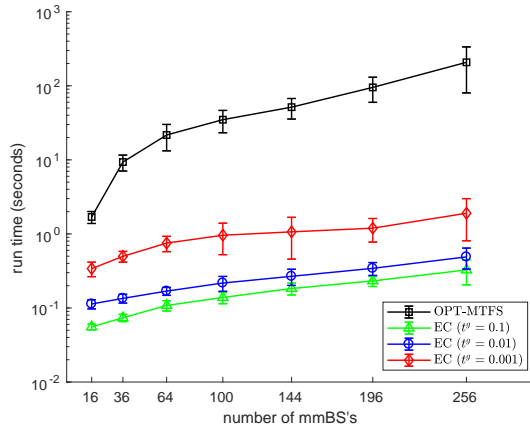
(b) run time

Fig. 6. Performance of the optimal MTFS algorithm (OPT-MTFS) and the EC-based approximation algorithm (EC) under the condition of single-RF-chain mmBSs. The curves show the average performance, and the error bars show $\pm$ standard deviation.
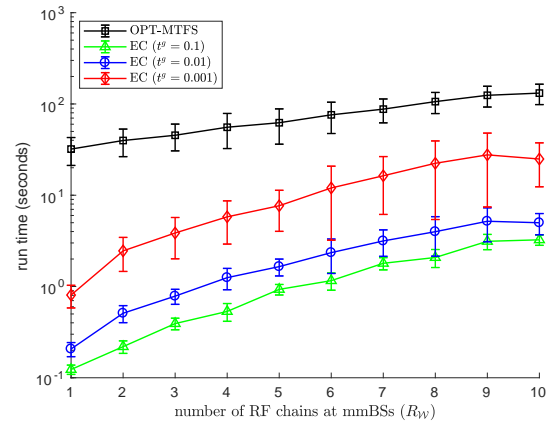

(a) max-min throughput θ

(b) run time

Fig. 7. Performance comparison for different number of RF chains at mmBSs. The curves show the average performance, and the error bars show $\pm$ standard deviation.

$t^g = 0.001$, we achieve 85% to 90% of the optimal max-min throughput while using 2% to 20% of the time.

IX. Conclusions

The paper presents an optimal joint routing and scheduling method—*schedule-oriented optimization* for mmWave cellular networks based on matching theory. It can solve any problem that can be formulated as a linear program whose variables are link times and QoS metrics. The method is demonstrated to be efficient in practice, capable of solving the maximum throughput fair scheduling (MTFS) problem within a few minutes for over 200 mmBSs. For better runtime efficiency, an edge-coloring based approximation algorithm is presented, which runs 5 to 100 times faster than the optimal algorithm while achieving over 80% of the optimal performance. In summary, the proposed optimal and approximation algorithms are highly practical for mmWave cellular networks.

Appendix

A. Proof of Lemma 1

Proof. We first prove part (2). Let a feasible schedule $\mathcal{S}$ consist of $N \geq 1$ slots. Because the total length of all slots is no more than 1, we have

$$\sum_{i=1}^{N} t_i \leq 1 \tag{10}$$

9

In [14], Edmonds' matching polyhedron theorem states that all matchings in a graph G are one-to-one mapped to the vertices of the matching polyhedron Q described by (11) (each vertex of Q has elements of either 0 or 1, where $x_e = 1$ means that the edge e is in the matching).

$$\sum_{e \in \delta(v)} x_e \leq 1 \qquad \forall v \in \mathcal{V}, \tag{11a}$$

$$\sum_{e \in \mathcal{E}(\mathcal{O})} x_e \leq \left\lfloor \frac{|\mathcal{O}|}{2} \right\rfloor \qquad \forall \text{ odd set } \mathcal{O} \subseteq \mathcal{V}, \tag{11b}$$

$$x_e \geq 0 \qquad \forall e \in \mathcal{E}. \tag{11c}$$

Since in each slot, the set of scheduled links is a matching in G, we further define $x_e^i = 1$ if the link e is active in i-th slot; otherwise, $x_e^i = 0$. Then for a given index i, each variable $x_e^i, e \in \mathcal{E}$ satisfies (11) when x_e is replaced with x_e^i. Combining (10) and (11), we have

$$\sum_{i=1}^{N} t_i \sum_{e \in \delta(v)} x_e^i \leq \sum_{i=1}^{N} t_i, \tag{12a}$$

$$\sum_{i=1}^{N} t_i \sum_{i \in \mathcal{E}(\mathcal{O})} x_e^i \leq \sum_{i=1}^{N} \left\lfloor \frac{|\mathcal{O}|}{2} \right\rfloor t_i. \tag{12b}$$

Since the link time $t_e = \sum_{i=1}^{N} t_i x_e^i \geq 0$, this implies that (12a) and (12b) are equivalent to (1a) and (1b), respectively. Thus, each feasible link time vector is a point in P.

We now prove part (1). Since the schedule polyhedron P is the same as the matching polyhedron Q, each vertex of P is a matching in G. Suppose that all the vertices of P are $\boldsymbol{x}_1, ..., \boldsymbol{x}_K$, where K is some positive integer. Since P is a convex set, it means by definition that each point $\boldsymbol{t} \in P$ can be expressed by a convex combination of the vertices of P:

$$\boldsymbol{t} = \sum_{k=1}^{K} \alpha_k \boldsymbol{x}_k,$$

where $\alpha_k \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$. This can be interpreted as follows: a point $\boldsymbol{t} \in P$ corresponds to a feasible schedule $\mathcal{S}$ of unit length. $\mathcal{S}$ has K slots and the length of the k-th slot is α_k. Here, the links correspond to $\boldsymbol{x}_k$ is a matching, and they are scheduled in the k-th slot. Thus, we have proved that any $\boldsymbol{t} \in P$ is feasible. □

B. Proof of Theorem 1

The proof applies the technique used in [18] to prove that fractional edge coloring can be solved in polynomial time by the ellipsoid algorithm. Specifically, a linear program is solvable in polynomial time if the separation problem of its dual problem can be solved in polynomial time. The separation problem of a linear program J is to determine whether a given solution satisfies all constraints of J or a violated constraint is identified. If we can solve both linear programs of (2) and (3) in polynomial time, then we can solve the MTFS problem in polynomial time.

We first prove that (2) can be solved in polynomial time. The dual of (2) is

$$\text{maximize} \qquad q \tag{13a}$$

$$\text{subject to} \quad \boldsymbol{p}^\mathsf{T} \mathsf{A}^{\mathcal{W}} + q \boldsymbol{1}^\mathsf{T} \leq \boldsymbol{0}^\mathsf{T} \tag{13b}$$

$$\boldsymbol{p}^\mathsf{T} \boldsymbol{1} = 1 \tag{13c}$$

$$\boldsymbol{p} \geq \boldsymbol{0}. \tag{13d}$$

Given a solution $(\boldsymbol{p}, q)$, (13c) and (13d) can be checked in polynomial time, since the total number of constraints in (13c) and (13d) is $W + 1$ and $\boldsymbol{p}$ contains W elements.

To check (13b), we use the polynomial maximum weighted matching algorithm [14]. A constraint in (13b) is of the form $\boldsymbol{p}^\mathsf{T} \boldsymbol{a}_k^{\mathcal{W}} \leq -q$, where $\boldsymbol{a}_k^{\mathcal{W}}$ is the k-th column of $\mathsf{A}^{\mathcal{W}}$ ($\boldsymbol{a}_k^{\mathcal{W}}$ corresponds to a matching). We set the weights $w_{(v_i, v_j)}$ to the links (v_i, v_j) such that

$$w_{(v_i, v_j)} = \begin{cases} c_{(v_i, v_j)}(p_j - p_i) & \text{if } v_i, v_j \in \mathcal{W} \\ c_{(v_i, v_j)} p_j & \text{otherwise } v_i \in \mathcal{R}, v_j \in \mathcal{W}. \end{cases} \tag{14}$$

Then we perform maximum weighted matching on G. If the weight of the maximum weighted matching satisfies $w \leq -q$, then $(\boldsymbol{p}, q)$ satisfies (13b). Otherwise the maximum weighted matching gives a violated constraint.

According to Theorem. 3.10 in [26], for a linear program J, if we can solve the separation problem of its dual J^* in polynomial time, then we can solve both J and J^* in polynomial time with the ellipsoid algorithm. This proves that (2) can be solved in polynomial time.

Similarly, we next prove that (3) can be solved in polynomial time. The dual of (3) is

$$\text{maximize} \qquad \theta \boldsymbol{p}^\mathsf{T} \boldsymbol{1} + q \tag{15a}$$

$$\text{subject to} \quad \boldsymbol{p}^\mathsf{T} \mathsf{A}^{\mathcal{W}} + q \boldsymbol{1}^\mathsf{T} \leq -\boldsymbol{c}^\mathsf{T} \tag{15b}$$

$$\boldsymbol{p} \geq \boldsymbol{0}. \tag{15c}$$

Given a tuple $(\boldsymbol{p}, q)$, we set the weights $w_{(v_i, v_j)}$ to the links (v_i, v_j) such that

$$w_{(v_i, v_j)} = \begin{cases} c_{(v_i, v_j)}(p_j - p_i) & \text{if } v_i, v_j \in \mathcal{W} \\ c_{(v_i, v_j)}(p_j + 1) & \text{otherwise } v_i \in \mathcal{R}, v_j \in \mathcal{W}. \end{cases} \tag{16}$$

Then we perform maximum weighted matching on G. Depending on whether the weight of the maximum weighted matching satisfies $w \leq -q$, the constraints of (15b) are satisfied or a violated one is identified. With the same argument as above, (3) can be solved in polynomial time. This complete the proof that the MTFS problem can be solved in polynomial time with the ellipsoid method.

REFERENCES

[1] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[2] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," in *Proceedings of the IEEE*, 2014.

[3] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2196–2211, 2015.

[4] A. Ghosh *et al.*, "Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1152–1163, 2014.

[5] Y. Niu *et al.*, "Exploiting device-to-device communications in joint scheduling of access and backhaul for mmwave small cells," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2052–2069, 2015.

[6] Y. Zhu *et al.*, "QoS-aware scheduling for small cell millimeter wave mesh backhaul," in *2016 IEEE ICC*, 2016, pp. 1–6.

[7] W. Feng *et al.*, "Millimetre-wave backhaul for 5g networks: Challenges and solutions," *Sensors*, vol. 16, no. 6, p. 892, 2016.

[8] Y. Li *et al.*, "A joint scheduling and resource allocation scheme for millimeter wave heterogeneous networks," in *WCNC*, 2017, pp. 1–6.

[9] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[10] P. K. Huang, X. Lin, and C. C. Wang, "A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 495–508, 2013.

[11] V. Angelakis *et al.*, "Minimum-time link scheduling for emptying wireless systems: Solution characterization and algorithmic framework," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1083–1100, 2014.

[12] S. Hariharan and N. B. Shroff, "On sample-path optimal dynamic scheduling for sum-queue minimization in trees under the k-hop interference model," in *IEEE INFOCOM*, 2012, pp. 999–1007.

[13] B. Ji, G. R. Gupta, and Y. Sang, "Node-based service-balanced scheduling for provably guaranteed throughput and evacuation time performance," in *IEEE INFOCOM*, 2016.

[14] J. Edmonds, "Maximum matching and a polyhedron with 0, 1 vertices," *J. of Res. the Nat. Bureau of Standards*, vol. 69 B, pp. 125–130, 1965.

[15] J. Tang, G. Xue, and W. Zhang, "Maximum throughput and fair bandwidth allocation in multi-channel wireless mesh networks," in *IEEE INFOCOM*, 2006, pp. 1–10.

[16] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *IEEE INFOCOM*, vol. 2, 2002, pp. 763–772.

[17] L. Khachiyan, "Polynomial algorithms in linear programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53 – 72, 1980.

[18] G. L. Nemhauser and S. Park, "A polyhedral approach to edge coloring," *Oper. Res. Lett.*, vol. 10, no. 6, pp. 315–322, 1991.

[19] G. B. Dantzig *et al.*, "The generalized simplex method for minimizing a linear form under linear inequality restraints," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 183–195, 1955.

[20] S.-i. Nakano, X. Zhou, and T. Nishizeki, *Edge-coloring algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 172–183.

[21] I. Holyer, "The NP-completeness of edge-coloring," *SIAM Journal on Computing*, vol. 10, no. 4, pp. 718–720, 1981.

[22] H. J. Karloff and D. B. Shmoys, "Efficient parallel algorithms for edge coloring problems," *J. Algorithms*, vol. 8, no. 1, pp. 39–52, 1987.

[23] R. Cole, K. Ost, and S. Schirra, "Edge-Coloring Bipartite Multigraphs in $O(E \log D)$ Time," *Combinatorica*, vol. 21, no. 1, pp. 5–12, 2001.

[24] M. R. Akdeniz *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, 2014.

[25] V. Kolmogorov, "Blossom V: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, 2009.

[26] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981.